Traffic Camera Dangerous Driver Detection TCD3™

Contextually Aware Heuristic Feature & OFA Density-Based Computer Vision with Movement Machine Learning Analysis of Live Streaming Traffic Camera Footage to Identify Anomalous & Dangerous Driving

Vidur Prasad | Dayton Regional STEM School | Dayton, OH

Outline

Motivation

>Summer Internship Project in Computer Science
- >Sensor Data Processing
- >Machine Learning

>Drunk/Distracted Driving is a serious problem
- >Personal Connection
- >Distracted Driving
  - >660,000 distracted drivers
  - >More than $50,000,000 per year
- >Drunk Driving
  - >10.3 Million Drunk Drivers
  - >Requires human-in-the-loop
  - >More than $40,000,000 per year

Technical Goal

> Input
- > Receive real time traffic camera data from surveillance, red light and traffic cameras

> Process data in real time

> Apply Computer Vision techniques to detect cars

> Create Intelligent System that gets smarter to build an accurate model of normal driving for anomaly detection

> *Test for low quality footage*

> *Optimize for high quality footage*

> Output
- > 10 second video clips
  - > Sent to Police Car's Laptops
  - > Showing video of suspected distracted drivers

Language & Environment

> C++
- > Low-level access
- > Matrix manipulation

> Prototyped with Octave
- > 2015 Project

>Redesigned with OpenCV
- >Developed by Intel
- >Large open-source community
- >Low level access compared to other libraries

Computer Vision

> Heuristic
- > Series of practical steps to effectively analyze images

> Background Subtraction

> Background Generation
- > Statistical Techniques

> Optical Flow Techniques

> Sliding Window Neighbor Detector
- > Refining Data

> Voting system to get clean detects

> Return center point guess for cars for all methods

> ~.59 Seconds per Frame

> Accuracy of 92%

Background Model

> Background Subtraction
- > Find difference in frames

> Update models to deal with moving sunlight

> Background Median Model
- > Median of Pixel Values
- > Quick approach
  - > ~.14 seconds per frame
- > Inaccurate
  - > 57% accuracy

> Gaussian Mixture Model

> MOG Gaussian Approach
- > Lighting Adjustments
- > Shadow Removal
- > Regular Movement

> ViBe
- > Implements Probability Density Function
- > Initial Training Period of 500 Frames
- > Actively Updating
- > Highest accuracy of methods

Optical Flow Analysis

> Quantify motion in video

> Determine the delta of pixels frame over frame

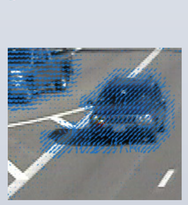
> Farneback Model

> Vectors returned quantifying movement

> Irregular Object Detection

> Future use

> Supplement Background
- > Diverse car colors have no effect
- > Even small cars are picked up by OFA
  - > OFA can detect even 1 pixel movement



Sliding Window Neighbor Detector

> Morphology is Industry Standard
- > Used to Refine Binary Image

> Address Noise

> Fix Errors in Binary Image

> Recursive Definition

> Density analysis

> Process
- > Create Window on Sensor Output
- > Slide Window across Binary Image
  - > If Window is filled to threshold, then completely fill Window
  - > If Window is not filled to threshold, completely empty Window
- > Increase size of window for multiple passes

> Increases speed over industry by 62%

> Reduces inaccurate detects by 47%

Machine Learning

> Historical Analysis

> Active Analysis
- > "Duck on the road"

> Matching Algorithm
- > K Means

> Learning Area Sector Model

> Car Movement Properties

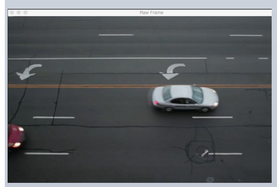
> System Paradigm
- > Supervised Algorithms
- > Unsupervised Nature

> Require violations in more than one properties

> ~.43 Seconds per Frame

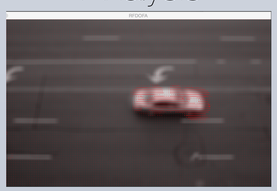
> Accuracy of 83%

Raw Frame

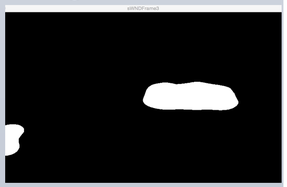


Background Image

Optical Flow Analysis

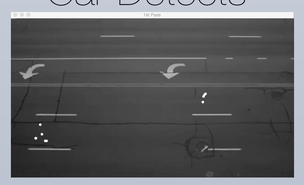


Sliding Window Neighbor Detector

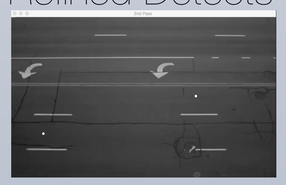


Final Detection

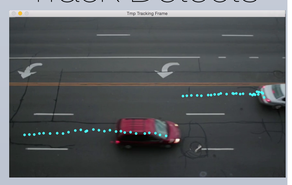
Car Detects



Refined Detects

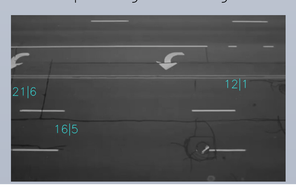


Track Detects

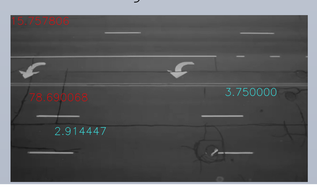


Learning Area Sector Model

Property Analysis



Anomaly Detection



Learning Area Sector Model

> Learn area properties
- > Complicated lanes

> Develop sectors for discrete paths

> Measure deviance from lane
- > Learn average displacement

> Measure average speed at points

> Measure average acceleration at points

> Measure angle at points

> Utilize safety factor to determine acceptable range

> Open Data Structure
- > API
- > Adapts to properties

> Track lane entry departure
- > Swerving

Property Learning

> Learn individual car properties for LASM points

> Hard Threshold for Speed Limits

> Learned Threshold

> Flexibility
- > Alter allowed error with safety

> Tag with weather and time
- > Adjust models for accuracy

> Sleep when speed drops below threshold

> Learned Properties
- > Speed
- > Acceleration
- > Direction
- > Angle

> 2 week training period

Distributed Computing

> 1.02 seconds per frame

> Multi-threading critical for performance
- > Computer Vision Processes
- > Machine Learning Processes
- > 8 threads

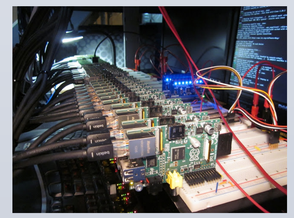
> CPU GPU Paradigm

> Master/Slave configuration used
- > Decentralized Management

> Raspberry Pi
- > Fully functioning computer
- > Cheap $30 Units

> Scalability
- > Reduces cost

> Risks associated with non-sequential approach



Future Plans

> Complete partnership with Dayton Law Enforcement
- > Footprint Integration
- > Security vs. Privacy Issues

> Run time must be lowered
- > Spark System

> Continue testing complete system with high quality dataset

> Uses existing visual sensing infrastructures

> Train system with Haar Classifier to increase future accuracy

References

> CDC. (n.d.). *Impaired Driving: Get the Facts.* Retrieved from Centers for Disease Control and Prevention: http://www.cdc.gov/Motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html

> Droogenbroeck, Marc Van. "ViBe: A Disruptive Method for Background Subtraction - Van Droogenbroeck Marc." *ViBe: A Disruptive Method for Background Subtraction - Van Droogenbroeck Marc.* ViBe, n.d. Web. (n.d)

> Foschi, Patricia G. "Feature Extraction for Content-Based Image Retrieval." (n.d.): n. pag. *Feature Extraction for Image Mining.* San Francisco State University. Web

> "Install Eclipse for C++ Development on OS X Mountain Lion." *Install Eclipse for C++ Development on OS X Mountain Lion.* Princeton, n.d. Web. 07 Apr. 2016.

> Kokash, N. *An Introduction to Heuristic Algorithms.* PSU.

> Jain. "Edge Detection." (n.d.): n. pag. *Edge Detection.* University of Nevada Reno. Web.

> Mathworks . *Morphology Fundamentals: Dilation and Erosion.* Mathworks.

> Mothers Against Drunk Driving. (2014). *Drunk Driving Statistics.* Retrieved 5 25, 2014, from MADD: http://www.madd.org/drunk-driving/about/drunk-driving-statistics.html

> Reynolds, Douglas. "Gaussian Mixture Models." (n.d.): n. pag. Print.

> Shi, Jianbo, and Carlo Tomasi. "Good Features to Track." *Good Features to Track* (1994): n. pag. Web.

> Shrivakshan, G. T. "A Comparison of Various Edge Detection Techniques Used in Image Processing." *International Journal of Computer Science Issues* 1st ser. 9.5 (2012): 269-76. *A Comparison of Various Edge Detection Techniques Used in Image Processing.* International Journal of Computer Science Issues, 1 Sept. 2012. Web.

> Sun, Min, and Srdjan Krstic. *Computer VisionToday's Outline* (n.d.): n. pag. *Optical Flow.* Princeton. Web.

> All Photos taken from Royalty-Free Sources

Thanks to Stack Exchange Community of Developers & the Open Source Community

Thanks to the University of Dayton Research Institute for fostering technical innovation

Thanks to Mr. Larrell Walters for Administrative Support

This project is dedicated to every innocent person that was killed in a drunk or distracted driving accident.